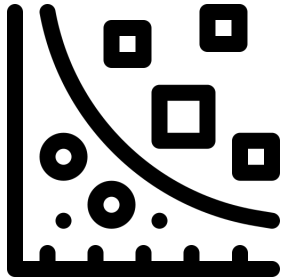
Ghouls, Goblins, and Ghosts... Boo!

Kaggle

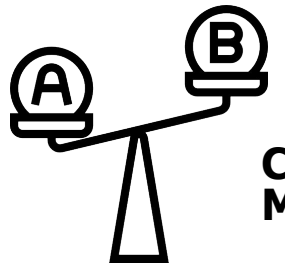
ПРЕЗЕНТАЦИЯ И РАСЧЕТЫ ПОДГОТОВЛЕНЫ:
Веркеенко Александр
Тимушкин Константин
Чернов Александр

Этапы работы с Dataset’ом

EDA



ДОБАВЛЕНИЕ НОВЫХ ПЕРЕМЕННЫХ ПОСЛЕ КЛАСТЕРИЗАЦИИ



СРАВНЕНИЕ МОДЕЛЕЙ

1 — 2 — 3

EDA

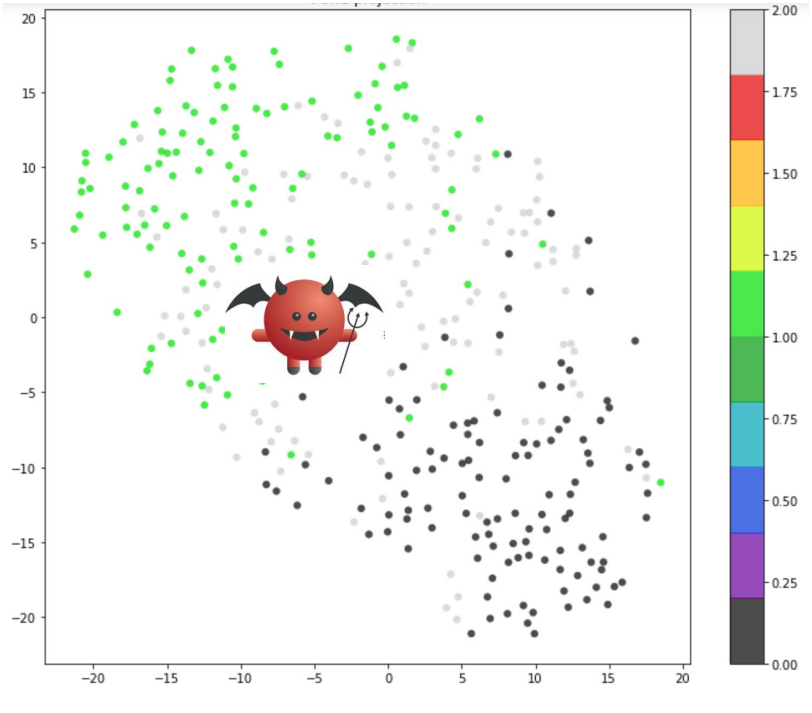
1. Проверили данные на пропуски
2. Визуально изучили взаимосвязь между типом монстра и его цветом **('Color')** => связь оказалась не очевидна, переменную оставили для дальнейших расчетов (ВСТАВИТЬ КАРТИНКУ С ГРАФИКАМИ)
3. Упрощаем эту переменную через One-Hot Encoding
4. Делаем перемножение признаков (посмотреть, как сделал чувак)
5. Попробуем сделать кластеризацию и добавить принадлежность к кластеру в качестве переменной

ДОБАВИЛИ НОВЫЕ ПРИЗНАКИ В НАДЕЖДЕ УВЕЛИЧИТЬ КАЧЕСТВО ПРЕДСКАЗАНИЯ

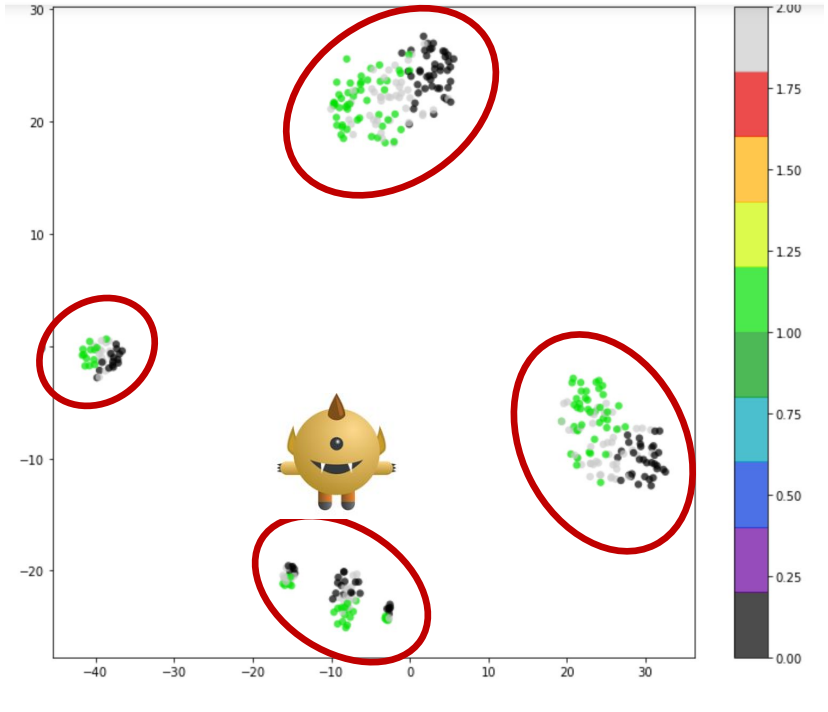
Добавление переменных в качестве кластера (1)

Сделали t-SNE на данных с OHE и без OHE

T-SNE с OHE



T-SNE без OHE

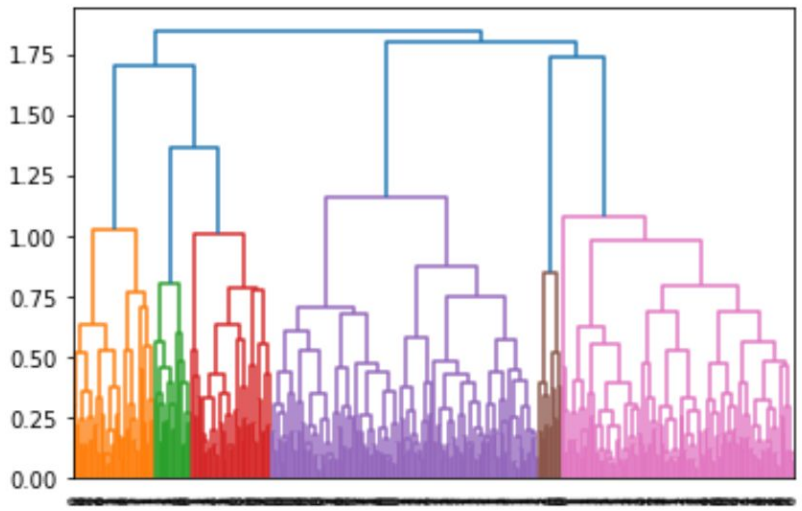


Визуально, на данных с OHE группы монстров сильно удалены друг от друга, но внутри каждой группы классы сильно перемешаны **=> OHE лучше?**

Добавление переменных в качестве кластера (2)

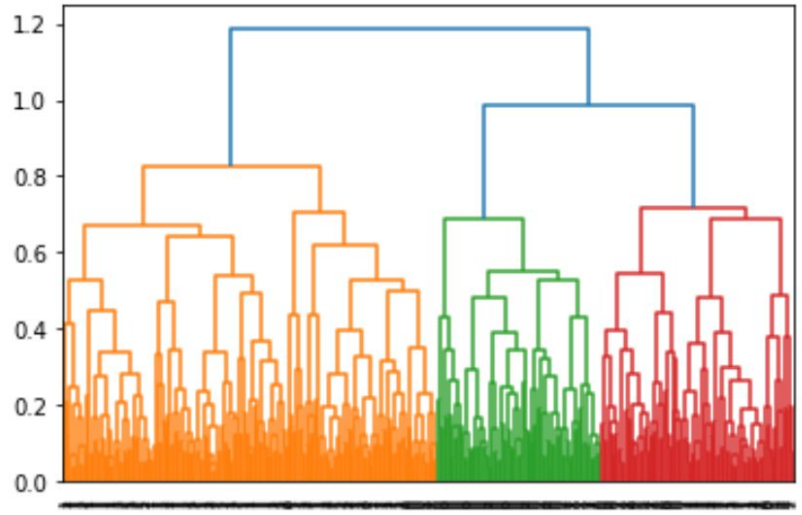
Сделали Дендрограмму на данных с OHE и без OHE

Дендрограмма с OHE



Wall time: 14.4 s

Дендрограмма без OHE



Wall time: 13.6 s

Дендрограмма без OHE хорошо подсвечивает 3 класса **=> берем на следующий шаг данные без OHE и без 'Color'**

Добавлен[illegible]е переменных в качестве кластера (3)

Кластеризация на train по 3-м моделям: Spectral, Agglomerative, K-Means

Spectral		Agglomerative		K-Means		OHE
Homogeneity:	0.273	Homogeneity:	0.365	Homogeneity:	**0.402**	0.004
Completeness:	0.274	Completeness:	0.367	Completeness:	**0.407**	0.004
V-measure:	0.274	V-measure:	0.366	V-measure:	**0.404**	0.004
Adjusted Rand-Index:	0.288	Adjusted Rand-Index:	0.385	Adjusted Rand-Index:	**0.431**	0.001
Adjusted mutual info score:	0.270	Adjusted mutual info score:	0.363	Adjusted mutual info score:	**0.401**	0.001
Silhouette Coefficient:	0.199	Silhouette Coefficient:	0.223	Silhouette Coefficient:	**0.221**	0.563

K-Means дает более высокие внешние + внутренние оценки, но Agglomerative очень близко какой вариант лучше? **GO Deeper!**

Добавление переменных в качестве кластера (4)

Кластеризация на train по 3-м моделям:
Spectral, Agglomerative, K-Means

Spectral

Кластеры:	1	2	3
GHOST	14.5	**83.8**	1.7
GHOUL	24.0	10.1	**65.9**
GOBLIN	**52.8**	21.6	25.6

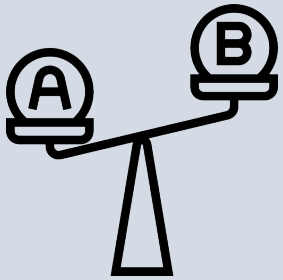
Agglomerative

Кластеры:	1	2	3
GHOST	7.7	**90.6**	1.7
GHOUL	22.5	4.7	**72.9**
GOBLIN	**54.4**	16.8	16.8

K-Means

Кластеры:	1	2	3
GHOST	10.3	1.7	**88**
GHOUL	14.0	**82.9**	3.1
GOBLIN	**55.2**	30.4	14.4

K-Means дает (1) более высокие внешние + внутренние оценки и (2) лучше разбивает вурдалаков на классы **=>**
берем K-Means на следующий шаг



СРАВНЕНИЕ МОДЕЛЕЙ

Модель	TRAIN	TEST	KAGGLE
Качество модели без работы с данными (что за модель, кстати?)	**0.72**	**0.72**	**0.72**
Random Forrest без кластеров на кластерах	**0.72** **0.75**	**0.72** **0.75**	**0.72** **0.75**
CATBOOST без кластеров на кластерах	**0.72** **0.75**	**0.72** **0.75**	**0.72** **0.75**
LinearSVC без кластеров на кластерах	**0.72** **0.75**	**0.72** **0.75**	**0.72** **0.75**
KNN без кластеров на кластерах	**0.72** **0.75**	**0.72** **0.75**	**0.72** **0.75**